

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Ims et al.  
Serial No. : 09/754,891  
Filed : January 5, 2001  
Title : TECHNIQUE FOR AUTOMATED E-BUSINESS SERVICES  
Attorney Docket : RSW920000077US1 (IBM010PA)  
Examiner : Michael K. Botts  
Art Unit : 2176

Mail Stop Appeal Brief--Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

**APPEAL BRIEF**

A request for a one month extension accompanies this paper.

1. *Real Party in Interest*

The Real Party in Interest in the present appeal is IBM Corporation, the assignee of the subject application.

2. *Related Appeals and Interferences*

There are no related appeals or interferences.

3. *Status of the Claims*

Claims 1-28 and 30 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by a document entitled "e" speak Tutorial, Version: Beta 2.2, December, 1999 (hereinafter the "prior art document"). Claims 3, 5, 10, 12, 17, 19, 23-27, 28 and 30 are being appealed.

4. *Status of Amendments*

First and second Amendments After Final were filed respectively on October 4, 2006 and November 1, 2006. Those papers were not entered by the Primary Examiner. On December 12,

2006, applicants filed a “Petition to the Director of the USPTO from Refusal of the Primary Examiner to Enter an Amendment After Final Rejection and after the Filing of a Notice of Appeal.” In the Amendment After Final dated November 1<sup>st</sup>, applicants deleted claims 3, 10 and 17 and added the limitations from those claims to independent claims 1, 8 and 15. Limitations similar to those added to claims 1, 8 and 15 were added to independent claim 22. Further, the language “configured for” was removed from claims 1, 2, 6 and 7. Supervisory Patent Examiner Heather R. Herndon indicated that the Amendment After Final Rejection dated November 1, 2006 changed the scope of the claims and, hence, would require further search and consideration.

5. *Summary of Claimed Subject Matter*

Set out below are independent and dependent claims annotated with paragraphs, line numbers and Figures from corresponding Published Application US 2002/0091533 A1.

Independent claim 1 recites:

A computer program product for automated e-business services, the computer program product embodied on one or more computer-readable media of a first computing system and comprising:

computer-readable program code configured for reading a specification of an e-business service (In corresponding Published Application US 2002/0091533 A1, see “Service definition scripts” in paragraph 73 and “the basic element of the XML service script definition is a ‘Service Unit,’” in paragraph 74.); and

computer-readable program code configured for processing the specification to carry out the e-business service, further comprising:

computer-readable program code configured for receiving at least one input document for the e-business service (“In the example flows illustrated in Fig. 5, a business initially receives two input XML documents \*\*\*.” See paragraph 72 of corresponding Published Application US 2002/0091533 A1.); and

computer-readable program code configured for performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create one or more new documents, according to operating actions that may be provided in the specification (“The actions element [of a “ServiceUnit”] describes the data transformations and business logic processing to be performed.” See Fig. 6 and paragraph 74 of corresponding Published Application US 2002/0091533 A1.).

Independent claims 8 and 15 recite substantially similar subject matter.

Dependent claim 3 recites:

The computer program product according to Claim 1, wherein the specification and the input documents are encoded in a structured markup language (In corresponding Published Application US 2002/0091533 A1, see “Service definition scripts” in paragraph 73 and “the basic element of the XML service script definition is a ‘Service Unit,’” in paragraph 74; “In the example flows illustrated in Fig. 5, a business initially receives two input XML documents \*\*\*.” See paragraph 72 of corresponding Published Application US 2002/0091533 A1; and “A structured markup language, which for discussion of the preferred embodiment is assumed to be XML \*\*\*.” See paragraph 60 of corresponding Published Application US 2002/0091533 A1.).

Dependent claims 10 and 17 recite substantially similar subject matter.

Independent claim 23 recites:

A method of defining e-business process and data interactions, further comprising:

defining data inputs to be used by an e-business service (“[I]nput XML documents such as the customer purchase orders \*\*\* contain the initial input data for processing customer orders \*\*\*.” See paragraph 71 of corresponding Published Application US 2002/0091533 A1.);

defining interactions to be carried out when operating the e-business service (“[E]ach trading partner prepares a set of ‘XML automation scripts’ for interactions with the other trading partners.” See paragraph 65 of corresponding Published Application US 2002/0091533 A1.);

specifying details of the data inputs in a structured markup language syntax (“In the example flows illustrated in Fig. 5, a business initially receives two input XML documents \*\*\*.” See paragraph 72 of corresponding Published Application US 2002/0091533 A1.);

specifying details of the interactions in the structured markup language syntax; and

creating at least one e-business service definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one e-business service definition document (“Each XML automation script \*\*\* will preferably specify the actions required to wait for an input data to be used \*\*\* (for example, actions that enable determining when the appropriate input data has been

received), and to act on the data inputs \*\*\*.” See paragraph 66 of corresponding Published Application US 2002/0091533 A1.).

Independent claim 25 recites:

A method of defining process and data interactions for an application described by a finite state machine, comprising:

defining data inputs to be used by the application “[I]nput XML documents such as the customer purchase orders \*\*\* contain the initial input data for processing customer orders \*\*\*.” See paragraph 71 of corresponding Published Application US 2002/0091533 A1.);

defining interactions to be carried out when operating the application (“[E]ach trading partner prepares a set of ‘XML automation scripts’ for interactions with the other trading partners.” See paragraph 65 of corresponding Published Application US 2002/0091533 A1.);

specifying details of the data inputs in a structured markup language syntax (“In the example flows illustrated in Fig. 5, a business initially receives two input XML documents \*\*\*.” See paragraph 72 of corresponding Published Application US 2002/0091533 A1.);

specifying details of the interactions in the structured markup language syntax; and

creating at least one application definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one application definition document (“Each XML automation script \*\*\* will preferably specify the actions required to wait for an input data to be used \*\*\* (for example, actions that enable determining when the appropriate input data has been received), and to act on the data inputs \*\*\*.” See paragraph 66 of corresponding Published Application US 2002/0091533 A1.).

Independent claim 27 recites:

A method for automating data and process interactions between a first application and one or more other applications, comprising:

providing at least one application definition document encoded in a structured markup language, wherein the application definition documents specify the interactions and at least one data input to be used in the interactions, and wherein details of the specified interactions and data inputs are specified in the structured markup language (“Each XML automation script \*\*\* will preferably specify the actions required to wait for an input data to be used \*\*\* (for example, actions that enable determining when the appropriate input data has been received), and to act on the data inputs \*\*\*.” See paragraph 66 of corresponding Published

Application US 2002/0091533 A1.); and

processing the application definition documents to carry out the data and process interactions in response to receiving at least one separate input document containing the at least one data input (“In the example flows illustrated in Fig. 5, a business initially receives two input XML documents \*\*\* from its partners. \*\*\* These documents are then combined and transformed into a single internal XML document by invoking a proper process (3). \*\*\* According to the present invention, this transformation operation may be described as an action for the data exchange engine to apply to an input XML document (7) if certain conditions are met, as defined in a service script XML document (8).” See paragraph 72 of corresponding Published Application US 2002/0091533 A1).

6. *Grounds of Rejection*

Whether claims 3, 5, 10, 12, 17, 19, 23-27, 28 and 30 are unpatentable under 35 U.S.C. § 102(b) as being anticipated by a document entitled “e”speak Tutorial, Version: Beta 2.2, December, 1999 (hereinafter the “prior art document”).

7. *Argument*

Rejection under 35 U.S.C. § 102(b) as being anticipated by a document entitled “e”speak Tutorial, Version: Beta 2.2, December, 1999.

Claims 3, 5, 10, 12, 17, 19, 27, 28 and 30

Independent claim 1 recites:

A computer program product for automated e-business services, the computer program product embodied on one or more computer-readable media of a first computing system and comprising:

computer-readable program code configured for reading a specification of an e-business service; and

computer-readable program code configured for processing the specification to carry out the e-business service, further comprising:

computer-readable program code configured for receiving at least one input document for the e-business service; and

computer-readable program code configured for performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create one or more new documents, according to operating actions that may be provided in the specification.

Dependent claim 3 recites:

The computer program product according to Claim 1, wherein the specification and the input documents are encoded in a structured markup language.

Claim sets 8, 10 and 15, 17 recite similar limitations.

Independent claim 27 recites:

A method for automating data and process interactions between a first application and one or more other applications, comprising:

providing at least one application definition document encoded in a structured markup language, wherein the application definition documents specify the interactions and at least one data input to be used in the interactions, and wherein details of the specified interactions and data inputs are specified in the structured markup language; and

processing the application definition documents to carry out the data and process interactions in response to receiving at least one separate input document containing the at least one data input.

The final Office Action states on page 6 regarding claim 3:

See, E-Speak, pages 76-77, teaching that the input documents are encoded in a structured markup language XML.

With regard to dependent claims 10 and 17, the Examiner relies upon the same statement set out above for claim 3. Please see page 8 of the final Office Action.

It is noted that in the Examiner's discussion of claim 3, he does not make reference to any portion of the prior art document for a teaching of providing a specification encoded in a structure markup language.

The Examiner states on page 4 of the final Office Action:

[T]he term "specification" is not defined in the specification. E-Speak defines "specification" as follows: "The events-service specification defines two interfaces. Namely, ListenerIntf: It defines the format of event notifications. DistributorIntf: If defines the format of publish and

subscribe requests.” See, E-Speak, page[s] 35-36. The E-Speak definition is consistent with the use of that term in this application, and, accordingly, the E-Speak definition of “specification” will be used in this Office Action.

However, the specification or any application definition document disclosed in the prior art document is not encoded in a structured markup language, as required by claims 3, 5, 10, 12, 17, and 19. Rather, the specification in the prior art document is written in Java programming language. Providing an e-business service specification or application definition document encoded in a structured markup language, i.e., in a universal exchange format, is believed to be advantageous as the specification or application definition document is independent of and not limited by a particular programming language such as Java.

With regard to claim 27, the final Office Action states on page 15:

(See, E-Speak, pages 75-77, teaching the providing and processing of documents in XML with specified interactions and data inputs in XML.)

The Examiner does not make reference to any portion of the prior art document for a teaching of providing an application definition document encoded in a structure markup language.

Accordingly, it is submitted that the prior art document does not anticipate the subject matter recited in claims 3, 5, 10, 12, 17, 19, 27, 28 and 30.

#### Claims 23-26

Independent claim 23 recites:

A method of defining e-business process and data interactions, further comprising:

defining data inputs to be used by an e-business service;  
defining interactions to be carried out when operating the e-business service;

specifying details of the data inputs in a structured markup language syntax;

specifying details of the interactions in the structured markup language syntax; and

creating at least one e-business service definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one e-business service definition document.

Independent claim 25 recites:

A method of defining process and data interactions for an application described by a finite state machine, comprising:

- defining data inputs to be used by the application;
- defining interactions to be carried out when operating the application;
- specifying details of the data inputs in a structured markup language syntax;
- specifying details of the interactions in the structured markup language syntax; and

creating at least one application definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one application definition document.

With regard to claim 23, the final Office Action states on page 11:

E-Speak contains code for reading and processing a specification of an e-business service. See E-Speak, pages 35-40.

A similar statement is made with regard to claim 25.

It is noted that the discussions in the final Office Action regarding claims 23 and 25 do not indicate that the prior art document provides specifying details of interactions to be carried out when operating an e-business service or an application described by a finite state machine in a structured markup language. In point of fact, nowhere does the prior art document disclose, teach or suggest specifying in structured markup language syntax details of interactions to be carried out when operating an e-business service or an application described by a finite state machine. Instead, it is believed that the prior art document discloses specifying in Java programming language details of interactions to be carried out when operating the E'speak

application. It is advantageous to specify in structured markup language syntax details of interactions to be carried out when operating an e-business service or an application such that the details of interactions are provided in a universal exchange format. Accordingly, it is submitted that claims 23-26 define patentably over the applied prior art.

### **CONCLUSION**

It is submitted that claims 3, 5, 10, 12, 17, 19, 23-27, 28 and 30 define patentably over the applied prior art. Accordingly, it is respectfully requested that the Board reverse the Examiner's final rejection of claims 3, 5, 10, 12, 17, 19, 23-27, 28 and 30.

Respectfully submitted,  
Stevens & Showalter, L.L.P.

By / Robert L. Showalter /

---

Robert L. Showalter  
Registration No. 33,579

January 26, 2007

7019 Corporate Way  
Dayton, OH 45459-4238  
Telephone: 937-438-6848  
Fax: 937-438-2124  
Email: [rshowalter@sspatlaw.com](mailto:rshowalter@sspatlaw.com)

8. *Claims Appendix*

1. A computer program product for automated e-business services, the computer program product embodied on one or more computer-readable media of a first computing system and comprising:

computer-readable program code configured for reading a specification of an e-business service; and

computer-readable program code configured for processing the specification to carry out the e-business service, further comprising:

computer-readable program code configured for receiving at least one input document for the e-business service; and

computer-readable program code configured for performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create one or more new documents, according to operating actions that may be provided in the specification.

2. The computer program product according to Claim 1, further comprising computer-readable program code configured for forwarding the other documents or the new documents to a computing system other than the first computing system.

3. The computer program product according to Claim 1, wherein the specification and the input documents are encoded in a structured markup language.

4. The computer program product according to Claim 1, wherein the other documents and the new documents are encoded in a structured markup language.

5. The computer program product according to Claim 3, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative

thereof.

6. The computer program product according to Claim 1, wherein the computer-readable program code configured for operating upon the input documents or the other documents further comprises:

computer-readable program code configured for invoking one or more software-implemented processes; and

computer-readable program code configured for coordinating results of the invocations.

7. The computer program product according to Claim 6, further comprising computer-readable program code configured for repetitively executing the computer-readable program code configured for processing, until reaching a final result of the e-business service, wherein the other documents, the new documents, or the coordinated results of the invocations now function as the input documents.

8. A system for automated e-business services, comprising:

means for reading a specification of an e-business service; and

means for processing the specification to carry out the e-business service, further comprising:

means for receiving at least one input document for the e-business service; and

means for performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create at least one new document, according to operating actions that may be provided in the specification.

9. The system according to Claim 8, further comprising means for forwarding the other documents or the new documents to a computing system other than the first computing system.

10. The system according to Claim 8, wherein the specification and the input documents are

encoded in a structured markup language.

11. The system according to Claim 8, wherein the other documents and the new documents are encoded in a structured markup language.

12. The system according to Claim 10, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative thereof.

13. The system according to Claim 8, wherein the means for operating upon the input documents or the other documents further comprises:

means for invoking at least one software-implemented process; and  
means for coordinating results of the invocations.

14. The system according to Claim 13, further comprising means for repetitively executing the means for processing, until reaching a final result of the e-business service, wherein the other documents, the new documents, or the coordinated results of the invocations now function as the input documents.

15. A method comprising:

reading a specification of an e-business service; and

processing the specification to carry out the e-business service, further comprising:

receiving at least one input document for the e-business service; and

performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create at least one new document, according to operating actions that may be provided in the specification.

16. The method according to Claim 15, further comprising forwarding the other documents or the new documents to a computing system other than the first computing system.

17. The method according to Claim 15, wherein the specification and the input documents are encoded in a structured markup language.
18. The method according to Claim 15, wherein the other documents and the new documents are encoded in a structured markup language.
19. The method according to Claim 17, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative thereof.
20. The method according to Claim 15, wherein operating upon the input documents or the other documents further comprises:
  - invoking at least one software-implemented process; and
  - coordinating results of the invocations.
21. The method according to Claim 20, further comprising repetitively executing processing the specification to carry out the e-business service, until reaching a final result of the e-business service, wherein the other documents, the new documents, or the coordinated results of the invocations now function as the input documents.
22. A method of conducting business by using automated e-business services, comprising:
  - reading a specification of an e-business service; and
  - processing the specification to carry out the e-business service, further comprising:
    - receiving at least one input document for the e-business service; and
    - performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the specification, and operating upon the input documents or the other documents to create at least one new document, according to operating actions that may be provided in the specification.

23. A method of defining e-business process and data interactions, further comprising:
  - defining data inputs to be used by an e-business service;
  - defining interactions to be carried out when operating the e-business service;
  - specifying details of the data inputs in a structured markup language syntax;
  - specifying details of the interactions in the structured markup language syntax; and
  - creating at least one e-business service definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one e-business service definition document.
24. The method according to Claim 23, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative thereof.
25. A method of defining process and data interactions for an application described by a finite state machine, comprising:
  - defining data inputs to be used by the application;
  - defining interactions to be carried out when operating the application;
  - specifying details of the data inputs in a structured markup language syntax;
  - specifying details of the interactions in the structured markup language syntax; and
  - creating at least one application definition document separate from at least one input document containing the data inputs wherein the specified details of the data inputs and the specified details of the interactions are recorded in the at least one application definition document.
26. The method according to Claim 25, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative thereof.
27. A method for automating data and process interactions between a first application and one or more other applications, comprising:

providing at least one application definition document encoded in a structured markup language, wherein the application definition documents specify the interactions and at least one data input to be used in the interactions, and wherein details of the specified interactions and data inputs are specified in the structured markup language; and

processing the application definition documents to carry out the data and process interactions in response to receiving at least one separate input document containing the at least one data input.

28. The method according to Claim 27, wherein processing the application definition documents further comprises:

receiving at least one input document to be used by the interactions; and

performing at least one of: transforming the input documents into other documents, according to transformation information that may be provided in the application definition documents, and operating upon the input documents or the other documents to create at least one new document, according to operating actions that may be provided in the application definition documents.

29. (Deleted)

30. The method according to Claim 27, wherein the structured markup language is a language known as “the Extensible Markup Language (XML)” or a derivative thereof.

9. *Evidence Appendix*

There is no additional evidence beyond the prior art document discussed above.

*10. Related Proceedings Appendix*

None